**Chapter 2 Recursion - Answers**

Top of Form

Bottom of Form

**Content**

**Answers**

**1. Given the function**

**int func(int num)  
    {  
        if (num == 2)  
            return 2;  
        else  
            return num \* func(num - 1);  
    }**

**a. Identify the base case in this function.**

The base case occurs when num == 2.

**b. Identify the recursive case in this function.**

The recursive case occurs when num != 2.

**c. What is output when this function is called using:**

**cout << func(5) << endl;**

   func(5) = 5 \* func(4)  
              = 5 \* 4 \* func(3)  
              = 5 \* 4 \* 3 \* func(2)  
              = 5 \* 4 \* 3 \* 2  
              = 120

The output is "120".

**d. The output statement in part c calls function func. How many times does func call itself as a result of the call in the output statement?**

The original call in the output statement generates 3 more calls to the function.

**e. There is a problem with this recursive function. Identify the problem.**

If the function is called with an argument that is smaller than 2, the base case will never be reached. The recursion will go on "forever". Actually, the program will abort when it runs out of memory to store the activation records.

**2. Section 2.4.1 of the textbook presents a recursive function to print the contents of a character array in reverse order. The function is:**

**/\*\* Write the characters in an array backward  
    @pre   anArray contains size = last - first + 1 characters,  
           where size > 0  
    @post  None  
    @param anArray - the array to write backward  
    @param first   - the index of the first character in the array   
    @param last    - the index of the last charater in the array  
   \*/  
   void writeArrayBackward(const char anArray[], int first, int last)  
   {  
       if (first <= last)  
       {  
          cout << anArray[last];                         // write the last character  
          writeArrayBackward(anArray, first, last - 1);  // write the rest of the array backward  
       }  
       // else nothing to write  
   } // end writeArrayBackward**

**Rewrite this function using 2 parameters - the array and the number of characters in the array.**

Here is a straight-forward rewrite:

   void writeArrayBackward(const char anArray[], int size)  
   {  
       if (size > 0)  
       {  
          cout << anArray[size - 1];                            // write the last character  
          writeArrayBackward(anArray, size - 1);    // write the rest of the array backward  
    }  
    // else nothing to write  
} // end writeArrayBackward

3**. Write a recursive function to print the contents of a character array character by charater in order. Use the original writeArrayBackward function from question #2 as a model.**

   void writeArray(const char anArray[], int first, int last)  
   {  
       if (first <= last)  
       {  
          cout << anArray[first];                        // write the first character  
          writeArrayBackward(anArray, first + 1, last);  // write the rest of the array in order  
       }  
       // else nothing to write  
   } // end writeArrayBackward